1. Write a program which allows the user to enter n numbers (n should be input by the user at the beginning of the program). Your program should print the sum of all of the even numbers that the user then enters.

2. 2520 is the smallest number that can be divided by each of the numbers from 1 to 10 without any remainder (IE: $2520\%1 = 0, 2520\%2 = 0, \ldots, 2520\%10 = 0$). Write a program that finds the smallest positive number that is evenly divisible by all of the numbers from 1 to 20?

3. Write a program that asks the user to input a number, n, and then print the sum of all numbers between 1 and n (inclusive) which are divisible by 4.

4. $n!$ is defined as $n * (n-1) * (n-2) * (n-3) * ... * (1)$. Thus, 4! is equal to $4 * 3 * 2 * 1 = 24$. Write a program that, given a number n from the user, prints out $n!$. $0! = 1$. *Warning: Factorial grows very quickly (factorially, in fact). You should only be able to compute low values of* $n!$ *before you experience an Integer overflow. This is okay.*

   (a) For a bonus, why does your program break when given values such as 15!?

5. Create a program that accepts an integer value from the user. Then print out this number in 8 bit binary (we can assume that all inputs will be $0 \leq n \leq 255$). If the user input the number 6, we would output: 00000110.

6. Create a program that takes a String, s, from the user, and creates a character array of the same contents. That is, given *"Bananas"*, you should create {'b','a','n','a','n','a','s'}.

7. Write a method signature for a method in the main class called *add* that takes three integers as an input, and returns the result of adding them.

---

8. Write a method signature for a method, called *printArray*, that takes an integer array as input and prints it.

---

9. Write a method signature for a method, called *quadFormula*, that takes three integers as input, and returns the quadratic formula of the three integers.

---

10. Write a method body for the following method, which will take two integer arrays as input and find their dot product. That is, given arrays a and b, you are finding: $\sum_i (a[i] * b[i])$:
public static int dotproduct(int[] a, int[] b){

}

Indicate what kind of error and if it's a compile time or runtime error, if any, the following will produce:

11. System.out.println(8 / 0);

---

12. String s = null;
    boolean b = s.equals("Hi");

---

13. System.Out.println("hello world!");

---

14. System.out.println("Hello!);

---

15. int[] array = $\{\{1, 2, 3\}, \{1, 2, 3\}\}$;

---

16. int[][] array = $\{\{1, 2, 3\}, 4\}$;

---

17. int[] array = $\{1, 2, 3, 4, 5\}$;
    array[5] = 6;

---

18. int size = 12;
    int[] array = new int[size];
    array[size - 1] = 100;

---

19. int[] array = new int[];

---

20. int[] array = new int[4];
    array = $\{1, 2, 3, 4, 5, 6\}$;

---

21. int j = "Hello world".length;

---

22. int i = "Hello world".indexOf('c');

---

23. int i = "Hello world".indexOf(12);

---

24. char c = "Hello world".indexOf('c');

---

25. char c = "Hello world".charAt(5);

---

26. char c = "Hello world".charAt(-1);

---

27. char c = "Hello world".charAt(11);

---

28. **Find five syntactical errors in the following snippet of code**:

```
for(int i = 0, i =< 6; i +)
        int value = i * 2;
        System.out.println(value)
}
```

29. **Find three logical errors in the following snippet of code, which is meant to print all of the values in an array, in order**:

```
int[] array = {1, 2, 3, 4, 5, 6};
for(int i = 0; i < array.length - 1; i --){
        int value = array[array.length - i];
        System.out.println(value);
}
```

30. Consider this code snippet:

```
int[] array = {4, 5, 6, 7, 8};
int mystery = 0;
for(int i = 0; i < array.length; i ++){
    mystery += array[i];
}
int mystery2 = array.length;
System.out.println((mystery * 1.0) / mystery2);
```

What is the output from the following code snippet?

How many times does the inner loop run?

In words, what does this snippet do?

31. Now consider this code snippet:

```
int alakazam = 0;
for(int i = 0; i < 8; i ++){
    for(int j = 0; j < i; j ++){
        alakazam += j;
    }
}
```

What is the output from the following code snippet?

How many times does the inner loop run?

In words, what does this snippet do?